

第八章 表格異動處理

SQL 語言除了提供資料查詢之外，也提供處理表格內容的相關命令，透過它們可以建立、刪除、變更表格，或插入、更新、移除、等等處理資料表內容。這些命令我們在第三章有提到的 DDL 與 DML 語法，但沒有介紹到，這裡補充說明。另外，對於表格的異動處理，並非只限於一個表格。一個異動處理也許會牽涉到多個表格的內容變更，這方面比較困難，爾後較進階的課程再介紹，本章只介紹針對一個表格的資料異動處理。

8-1 建立表格 – Create table

8-1-1 Create Table 語法

看得出來，Create Table 是建立資料表的命令，可以指定幾個欄位、每個欄位的資料格式、指定該表格主鍵、或外來鍵等等，命令格式如下：

```
Create Table 資料表名稱  
(欄位_1 資料型態 屬性,  
 欄位_2 資料型態 屬性,  
  ...,  
  Primary Key (欄位_1, 欄位_2)  
)
```

其中資料型態，常用有：

- 字元：CHAR(長度)，CHAR(8)。
- 整數：int(長度)，int(8)。
- 日期：

另外，屬性有：

- 不可空白：Not null

- Primary Key：指定主鍵，也許會包含兩個以上欄位。

8-1-2 範例研討：建立監護人表格

(A) 程式功能：Ex8_1

請建立一個學生監護人 (guardian)資料表，包含有：學號(student_ID)、監護人(guardian)、電話(tel)、地址(address)。

(B) 系統分析：

學號表示每一位學生的識別，它應該是這表格唯一不可重複的，我們就定它為主鍵，每個欄位都可採用字元格式，但地址的長度較長一點就好。

(C) 程式範例：

```
create table guardian (
    student_ID char(10) not null,
    guardian char(20),
    tel char(20),
    address char(50),
    primary key (student_ID)
);
create table guardian (
    student_ID char(10) not null,
    guardian char(20),
    tel char(20),
    address char(50),
    primary key (student_ID)
);
```

| 資料表 | 執行 | 記錄 | 型態 | 校對 | 大小 | 多餘 | |
|---------------|----|-----|--------|-----------------|-----------------|---------|---|
| all_course | | | 14 | MyISAM | utf8_unicode_ci | 3.6 KB | - |
| classes | | | 12 | MyISAM | utf8_unicode_ci | 3.4 KB | - |
| department | | | 11 | MyISAM | utf8_unicode_ci | 2.3 KB | - |
| guardian | | | 0 | MyISAM | utf8_unicode_ci | 1.0 KB | - |
| open_course | | | 20 | MyISAM | utf8_unicode_ci | 4.5 KB | - |
| select_course | | | 110 | MyISAM | utf8_unicode_ci | 5.1 KB | - |
| students | | | 55 | MyISAM | utf8_unicode_ci | 8.3 KB | - |
| teachers | | | 20 | MyISAM | utf8_unicode_ci | 5.0 KB | - |
| 8 資料表 | 總計 | 242 | MyISAM | utf8_unicode_ci | 33.2 KB | 0 Bytes | |

吾人亦可觀察 guardian 表格之結構如何：

| 欄位 | 型態 | 校對 | 屬性 | Null | 預設值 | 附加 | 執行 |
|-------------------------------------|----------|-----------------|----|------|------|----|----|
| <input type="checkbox"/> student_ID | char(10) | utf8_unicode_ci | | 否 | | | |
| <input type="checkbox"/> guardian | char(20) | utf8_unicode_ci | | 是 | NULL | | |
| <input type="checkbox"/> tel | char(20) | utf8_unicode_ci | | 是 | NULL | | |
| <input type="checkbox"/> address | char(50) | utf8_unicode_ci | | 是 | NULL | | |

↑ 全選 / 全部取消 選擇的資料表:

列印檢視 分析資料表結構

新增 1 個欄位 於資料表尾端 於資料表開頭 在 student_ID 之後 執行

| 索引: ⑦ | | | | 已使用空間 | | 資料列統計數值 | |
|---------|---------|----|----|------------|-----------|-------------|--------------------|
| 鍵名 | 型態 | 組別 | 執行 | 欄位 | 型態 | 使用 | 敘述 |
| PRIMARY | PRIMARY | 0 | | student_ID | 資料 | 0 Bytes | 格式 固定 |
| 新增 1 | 組索引欄 | 執行 | | | 索引鍵 INDEX | 1,024 Bytes | 校對 utf8_unicode_ci |
| | | | | | 總共 | 1,024 Bytes | 資料列列數 0 |

8-2 變更表格 – Alter Table

8-2-1 Alter Table 語法

當表格格式不滿足所需時，可以利用 Alter Table 變更型態，可針對現有表格作：增加欄位、修改欄位、刪除欄位，變更主鍵，以及定義屬性等等功能，格式如下：

Alter Table 資料表名稱

[Alter] [欄位] [資料型態][Null | Not null]

[Add | Drop] [欄位] [屬性][限制]

8-2-2 範例研討：增加關係欄位

(A) 程式功能：Ex8_2

請在監護人(guardian)資料表內增加關係 (relation) 欄位。

(B) 程式範例：

```
Alter Table guardian Add relation char(10);
```

吾人再觀察 guardian 表格之結構如何：

| 欄位 | 型態 | 校對 | 屬性 | Null | 預設值 | 附加 |
|-------------------|----------|-----------------|----|------|------|----|
| <u>student_ID</u> | char(10) | utf8_unicode_ci | | 否 | | |
| guardian | char(20) | utf8_unicode_ci | | 是 | NULL | |
| tel | char(20) | utf8_unicode_ci | | 是 | NULL | |
| address | char(50) | utf8_unicode_ci | | 是 | NULL | |
| relation | char(10) | utf8_unicode_ci | | 是 | NULL | |

8-3 插入資料 – Insert Into

8-3-1 Insert Into 語法

將資料新增到表格則利用 Insert 命令來達成，格式如下：

```
Insert Into 資料表名稱 <欄位串列>
Values (<資料串列>);
```

8-3-2 範例研討：輸入監護人資料

(A) 程式功能：Ex8_3

請輸入監護人資料，如下：

1. 學號：90211304、監護人：劉媽媽、電話：0923111223、地址：高雄市苓雅區 20 號、關係：母女。
2. 學號：90211305、監護人：張先生、電話：0923111277、地址：高雄市苓雅區 30 號、關係：夫妻。
3. 學號：90211306、監護人：劉爸爸、電話：0923111266、地址：高雄市苓雅區 40 號、關係：父子。

(B) 程式範例：

```
Insert Into guardian
  values("90211304", "陳媽媽", "0923111223", "高雄市苓雅區 20 號", "母女");
Insert Into guardian
  values("90211305", "張先生", "0923111277", "高雄市苓雅區 30 號", "夫妻");
Insert Into guardian
```

```
values("90211306", "劉爸爸", "0923111266", "高雄市苓雅區 40 號", "父子");
```

吾人再觀察 guardian 表格的內容如何：

```
Select *
From guardian;
```

| student_ID | guardian | tel | address | relation |
|------------|----------|------------|-------------|----------|
| 90211304 | 陳媽媽 | 0923111223 | 高雄市苓雅區 20 號 | 母女 |
| 90211305 | 張先生 | 0923111277 | 高雄市苓雅區 30 號 | 夫妻 |
| 90211306 | 劉爸爸 | 0923111266 | 高雄市苓雅區 40 號 | 父子 |

8-4 更新資料 – Update

8-4-1 Update 語法

新資料取代現有資料表的內容，稱之為『更新』(Update)，但資料表內有多筆資料，到底要更新哪一筆(或稱紀錄)或多筆資料同時更新，因此需要有一個 Where 敘述句來決定更新哪一筆或哪些資料。格式如下：

```
Update 資料表名稱
Set [<欄位_1>=<數值_1>, ...<欄位_2>=<數值_2>]
Where <條件敘述>;
```

上述語句中，只要某一筆滿足 Where 條件敘述，則更新它指定欄位的內容。

8-4-2 範例研討：變更監護人資料

(A) 程式功能：Ex8_4

請將學號：90211304 監護人資料中將關係(relation)改為『姊妹』。

(B) 程式範例：

```
Update guardian
Set relation = "姊妹"
Where student_ID = "90211304"
```

吾人再觀察 student_ID 資料內的關係(relation) 是否被變更：

```
Select *
From guardian
Where student_ID = "90211304";
```

| student_ID | guardian | tel | address | relation |
|------------|----------|------------|-------------|----------|
| 90211304 | 陳媽媽 | 0923111223 | 高雄市苓雅區 20 號 | 姊妹 |

8-5 刪除資料 – Delete From

8-5-1 Delete From 語法

刪除資料表中某一筆或多筆記錄則用 Delete 命令，但也須利用 Where 條件敘述來選取哪些紀錄要被刪除。格式如下：

```
Delete From 資料表名稱
Where <條件敘述>;
```

上述語句中，滿足 Where 條件敘述的紀錄，其內容將被刪除。

8-5-2 範例研討：刪除監護人資料

(A) 程式功能：Ex8_5

請將學號：90211304 監護人資料刪除。

(B) 程式範例：

```
Delete
From guardian
Where Student_ID = "90211304"
```

吾人再觀察 student_ID = 90211304 資料是否還在 guardian 表內：

```
Select *
From guardian;
```

| student_ID | guardian | tel | address | relation |
|------------|----------|------------|-------------|----------|
| 90211305 | 張先生 | 0923111277 | 高雄市苓雅區 30 號 | 夫妻 |
| 90211306 | 劉爸爸 | 0923111266 | 高雄市苓雅區 40 號 | 父子 |

8-6 刪除表格 – Drop Table

8-6-1 Drop Table 語法

當表格格式不需要時，可以利用 Drop Table 命令刪除它，當然它裡面資料也會一併被刪除掉，格式如下：

```
Drop Table 資料表名稱 [Restrict | Cascade];
```

其中，Restrict 與 Cascade 表示資料表之間的關聯也會被刪除。

8-6-2 範例研討：刪除監護人表

(A) 程式功能：Ex8_6

請刪除監護人(guardian) 表格。

(B) 程式範例：

```
Drop Table guardian
```

吾人再觀察 course_db 內是否存在 guardian 表：

```
Show Tables;
```

| Tables_in_course_db |
|---------------------|
| all_course |
| classes |
| department |
| open_course |
| select_course |
| students |
| teachers |

8-7 表格異動整合運用

8-7-1 暫存資料表的運用

在第六章我們介紹了『子查詢』與『合併查詢』的整合運用，但發現其中最大困難是合併後的『動態資料表』，必須立即連接子查詢或與其他資料表（或動態資料表）合併，有時候真的很難連結。如果我們能產生暫時資料表，並將查詢後的動態資料表儲存起來，對於查詢語句就會比較好編寫，尤其在資料庫程式設計方面就會簡單許多。當查詢運作完成後，再將

這些暫存資料表刪除即可。

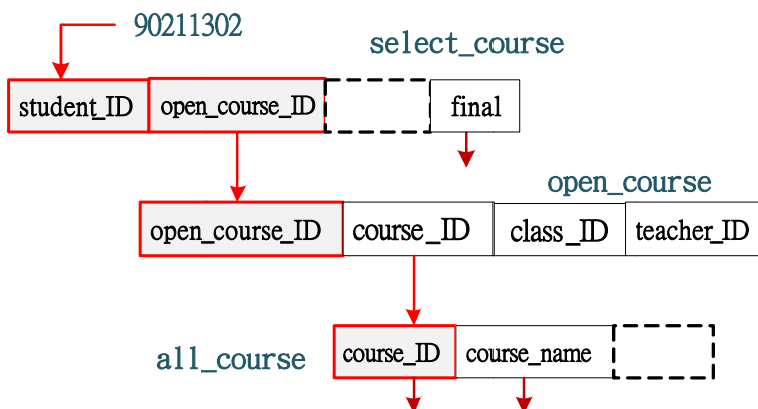
8-7-2 範例研討：查詢學生修課成績

(A) 程式功能：Ex8_7

請查詢 90211302 同學這學期修讀了幾門課，印出課程名稱與分數。

(B) 系統分析：

此題目在第七章(Ex7_2) 練習過，利用混合查詢的缺點是必須及時連結臨時表格，如果我們產生一些暫存資料表來承接臨時表格的結果，就能解決此問題。我們依照下圖資料的流程，一步一步將結果存入暫存資料表。



(C) 程式範例：

- **步驟 1:** 首先須由 select_course 表內查詢出該學生修了那些課，以及成績，並建立 temp1 = {open_course_ID, final}，將查詢結果存入其中。

```
Create Table temp1 AS
Select open_course_ID, final
From select_course
Where student_ID = "90211302";
Select *
From temp1;
```

| open_course_ID | final |
|----------------|-------|
| 1 | 86 |
| 4 | 88 |
| 6 | 87 |

- **步驟 2**：接著，將 open_course 與 temp1 表格合併，查詢出該生修課的課程編碼 (course_ID) 與成績，並產生 temp2 表將其結果存入，如下：

```
Create Table temp2 AS
Select course_ID, final
From temp1 as A, open_course as B
Where A.open_course_ID =B.open_course_ID;
Select *
From temp2;
```

| course_ID | final |
|-----------|-------|
| 90B252 | 86 |
| 90B253 | 88 |
| 90A900 | 87 |

- **步驟 3**：接著將 temp2 與 all_course 表格合併，查詢出課程名稱與修課成績。最後將 temp1 與 temp2 兩表格刪除，如下：

```
Select course_name, final
From temp2 as A, all_course as B
where A.course_ID = B.course_ID;
Drop table temp1;
Drop table temp2;
```

| course_name | final |
|-------------|-------|
| 資料庫管理系統 | 86 |
| 資料結構 | 88 |
| 數位內容概論 | 87 |

8-8 綜合練習

請利用暫時資料表與混合查詢之整合運用，製作下列題目：

- (A) 自我挑戰 PM8_1：請查詢『進四資二丙』這學期開了哪些課程，請依照課程名稱、必選修、學分數與授課教師之順序印出。
- (B) 自我挑戰 PM8_2：請查詢粘添壽老師這學期所開的課程，請依照班級名稱、課程名稱、必選修、學分數等順序印出。
- (C) 自我挑戰 PM8-3：請查詢資訊管理系這學期開了哪些課程，請依課程名稱、班級名稱、授課教師、必選修與學分數之順序印出。